This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-143 (Cancelled)

144. (Currently Amended) A method of execution-instruction delegation among multiple elemental processing resources of at least two different types, the multiple elemental processing resources comprising two or more elemental processing resources of a first type each sharing two or more elemental processing resources of a second type with ~~the~~ at least one other of the two or more processing resources of the first type, the processing resources of the first type being each being defined by a configuration whereby it will receive a non-delegated instruction from memory that it can execute or delegate for execution elsewhere, the processing resources of the second type each being defined by being configured to only receive and execute instructions that were delegated to them by processing resources of the first type and by being shared by at least two elemental processing resources of the first type, the method comprising:

obtaining an execution instruction, wherein the execution instruction is obtained at one of the elemental processing resources of the first type;

determining whether an operation-code within the execution instruction is incapable of being executed by the one elemental processing resource of the first type and thus should be delegated to one ~~of the shared~~ elemental processing resource of the second type that the one elemental processing resource of the first type shares in common with another elemental processing resource of the first type;

executing the execution instruction with the one elemental processing resource of the first type, if the operation-code within the execution instruction should not be delegated to any elemental processing resource of the second type; and

if the execution instruction should be delegated, routing the execution instruction to [[a]] the ~~shared~~ elemental processing resource of the second type~~, of the at least two of that type,~~ that is shared in common and is capable of executing the operation code.

145. (Original) The method of claim 144, wherein the method is completed within a single processing cycle.

Claims 146-154 (Cancelled)

155. (Original) The method of claim 144, wherein the operation-code indicates a type of resource on which to execute.

156. (Previously Presented) The method of claim 144, wherein at least one elemental processing resource of the first type is an originating processing resource.

157. (Previously Presented) The method of claim 144, wherein one or more of the elemental processing resources of the first type is an integer processing unit.

158. (Previously Presented) The method of claim 144, wherein one or more of the elemental processing resources of the second type is a mathematical processing unit.

159 (Cancelled)

160. (Previously Presented) The method of claim 144, wherein one or more of the elemental processing resources of the second type is a vector processing unit.

Claims 161-163 (Cancelled)

164. (Previously Presented) The method of claim 144, wherein one or more of the elemental processing resources of the second type is an execution-instruction processing cache.

165. (Original) The method of claim 144, further comprising routing the execution instruction through an execution-instruction signal router.

166 (Cancelled)

167. (Currently Amended) The method of claim 144, wherein a first specific elemental processing resource executing a first individual thread may sleep while a second specific elemental processing resource executes delegated execution-instructions from the first individual thread.

168. (Previously Presented) The method of claim 144, wherein an execution-instruction signal causes various elemental processing resources dynamically to turn on and off to maintain a desired level of power draw while maximizing processing throughput.

169. (Currently Amended) The method of claim 144, further comprising generating an execution-instruction signal from at least one of the elemental processing resources, wherein the execution-instruction signal from the elemental processing resources themselves shuts off processing resources while idling.

170. (Currently Amended) The method of claim 144, further comprising generating an execution-instruction signal from at least one of the elemental processing resources, wherein an execution-instruction signal from processing resources themselves turn on processing resources when execution-instruction signal processing is required.

171. (Previously Presented) The method of claim 144, wherein the elemental processing resources are communicatively disposed on a same die.

172. (Previously Presented) The method of claim 171, wherein an execution-instruction signal router is on the same die with elemental processing resources.

Claims 173-1614 (Cancelled)

1615. (Currently Amended) A method of execution-instruction delegation among multiple interdependent processing resources, the processing resources comprising multiple elemental processing resources of (a) at least two IPU type processing resources and (b) two or more elemental processing resources of an other type comprising, in any combination, one or more of either an MPU type, an instruction delegating cache type, an encryption or decryption type or a vector type, the multiple elemental processing resources being configured to perform processing operations according instructions in an instruction set but being individually incapable of servicing at least one instruction in the instruction set, the elemental processing resources being configured such that a first and a second of the IPU type processing resources each share at least two of the other type elemental processing resources in common the method comprising:

receiving, from memory, one of multiple execution-instructions ~~from~~ of a thread at a first Instruction Processing Unit (IPU) of the at least two IPU type processing resources;

processing the first execution-instruction using the first IPU;

receiving an other execution-instruction ~~from~~ of the thread at the first IPU;

determining that the other execution-instruction from the thread can not be processed by the first IPU;

delegating the other execution-instruction from the thread to one of the two or more elemental processing resources of the other type that is shared between the first IPU and a second IPU of the at least two IPU type processing resources, the type of the processing resource other than the IPU type processing resource being such that it can process the other execution-instruction from the thread; ~~and~~

maintaining the thread in the first IPU in a sleep state until an indicator that the processing of the other execution-instruction from the thread by the other type processing resource is returned;

receiving an execution instruction of a different thread at the second IPU;

determining that the execution-instruction of the different thread can not be processed by the second IPU but can be processed by the other type elemental processing resource that is shared between the second IPU and the first IPU;

delegating the execution-instruction from the different thread to the other type elemental processing resource that is shared between the second IPU and the first IPU.